

Exercice 1 : Classe de Gestion d'Articles

Créez une classe `Article` avec les caractéristiques suivantes :

- Un constructeur qui accepte un titre et un contenu.
- Une méthode `afficher()` qui retourne une chaîne formatée contenant le titre et le contenu.

Correction :

```
class Article {
  constructor(titre, contenu) {
    this.titre = titre;
    this.contenu = contenu;
  }

  afficher() {
    return `Titre: ${this.titre}\nContenu: ${this.contenu}`;
  }
}

// Exemple d'utilisation
const article = new Article("Nouvelle découverte", "Un astéroïde passe près de la Terre.");
console.log(article.afficher());
```

Exercice 2 : Classe de Gestion d'Utilisateurs

Créez une classe `Utilisateur` qui :

- Prend un nom et un email dans son constructeur.
- Possède une méthode `afficherDetails()` qui affiche les informations de l'utilisateur.
- Ajoutez une méthode statique `afficherNombre()` qui affiche combien d'instances ont été créées.

Correction :

```
class Utilisateur {
  static compteur = 0;

  constructor(nom, email) {
    this.nom = nom;
    this.email = email;
    Utilisateur.compteur++;
  }

  afficherDetails() {
    return `Nom: ${this.nom}, Email: ${this.email}`;
  }

  static afficherNombre() {
    return `Nombre d'utilisateurs créés : ${Utilisateur.compteur}`;
  }
}
```

```
// Exemple d'utilisation
const user1 = new Utilisateur("Alice", "alice@example.com");
const user2 = new Utilisateur("Bob", "bob@example.com");

console.log(user1.afficherDetails());
console.log(Utilisateur.afficherNombre());
```

En JavaScript, une **méthode statique** est une méthode associée à une **classe** elle-même et non à une instance de la classe. Cela signifie qu'elle peut être appelée directement sur la classe, sans qu'il soit nécessaire de créer une instance.

Dans cet exemple, la méthode statique est :

```
static afficherNombre() {
    return `Nombre d'utilisateurs créés : ${Utilisateur.compteur}`;
}
```

Caractéristiques des Méthodes Statiques :

1. **Mots-clés `static`** : Utilisé pour définir une méthode statique ou une propriété statique.
2. **Accès via la classe** : Une méthode statique est appelée directement sur la classe, pas sur une instance.
 - **Exemple** : `Utilisateur.afficherNombre()`.
3. **Pas d'accès à `this` d'une instance** : Les méthodes statiques ne peuvent pas utiliser `this` pour accéder aux propriétés d'instance. Elles travaillent uniquement avec les propriétés/méthodes statiques de la classe.

Exercice 3 : Créer une Galerie Dynamique en JavaScript

Énoncé :

L'objectif de cet exercice est de créer une classe JavaScript nommée `Galerie` pour afficher dynamiquement des images dans une page web. Vous devez implémenter les fonctionnalités suivantes :

1. Ajouter des URL d'images à une galerie.
2. Afficher ces images dans une zone dédiée.
3. Régénérer la galerie pour refléter les modifications.



```
> galerie.ajouterImage('./images/tunisia4.jpg');
galerie.regenerer();
```



Suivez les étapes ci-dessous pour créer votre galerie.

Étapes :

1. Structure HTML

Créez une page HTML qui contient :

- Une balise `<div>` vide pour afficher les images.
- Une zone de script pour votre JavaScript.

2. Créer la Classe `Galerie`

- Ajoutez un constructeur qui accepte un conteneur HTML (`<div>`) comme argument. Ce conteneur servira à afficher les images.
- Déclarez une propriété `urlImages` pour stocker les URLs des images.

3. Ajouter des Méthodes à la Classe

- `ajouterImage(url)` : Cette méthode ajoute l'URL d'une image à la liste des images.
- `regenerer()` : Cette méthode :
 - Efface le contenu du conteneur (`innerHTML`).
 - Parcourt la liste des URLs et crée une balise `` pour chaque image, avec les styles spécifiés.

4. Utilisation de la Classe

- Instanciez la classe `Galerie` en passant un élément `<div>` de votre page comme conteneur.
- Ajoutez plusieurs images avec des URLs valides.
- Appelez la méthode `regenerer()` pour afficher les images.

Exercice à réaliser :

1. Ajoutez une nouvelle méthode `supprimerImage(url)` à la classe `Galerie`.

- Cette méthode doit supprimer une image spécifique en fonction de son URL.

2. Ajoutez un bouton sur la page pour effacer toutes les images de la galerie.

- Lorsque vous cliquez sur ce bouton, le contenu de la galerie doit être vidé.

Indications :

- **Manipulation DOM :** Utilisez `document.createElement` pour créer des éléments dynamiquement.
- **Styles :** Utilisez `style` pour personnaliser l'affichage des images.
- **Suppression d'éléments :** Vous pouvez utiliser `filter()` pour mettre à jour la liste des URLs.

Solution attendue :

L'étudiant doit produire une galerie interactive où :

1. Les images s'affichent en fonction des URLs ajoutées.
2. Il est possible de supprimer une image spécifique ou toutes les images.

Cela leur permettra de comprendre la manipulation du DOM et la création de classes en JavaScript.

Exercice 4 : Classe pour un Compteur

Créez une classe `Compteur` qui :

- Possède une propriété `valeur` initialisée à zéro.
- Fournit des méthodes `incrementer()` et `decrementer()`.
- Fournit une méthode `reset()` pour remettre le compteur à zéro.

Correction :

```
class Compteur {
  constructor() {
    this.valeur = 0;
  }

  incrementer() {
    this.valeur++;
  }

  decrementer() {
    this.valeur--;
  }

  reset() {
    this.valeur = 0;
  }
}
```

```
    afficher() {
        return `Valeur actuelle : ${this.valeur}`;
    }
}

// Exemple d'utilisation
const compteur = new Compteur();
compteur.increments();
compteur.increments();
console.log(compteur.afficher());
compteur.reset();
console.log(compteur.afficher());
```

Exercice 5 : Classe de Gestion de Livres

Créez une classe `Livre` avec les fonctionnalités suivantes :

- Un constructeur qui prend un titre, un auteur, et une année de publication.
- Une méthode `afficher()` qui retourne une description du livre.

Correction :

```
class Livre {
    constructor(titre, auteur, annee) {
        this.titre = titre;
        this.auteur = auteur;
        this.annee = annee;
    }

    afficher() {
        return `${this.titre}, écrit par ${this.auteur} en ${this.annee}.`;
    }
}

// Exemple d'utilisation
const livre = new Livre("1984", "George Orwell", 1949);
console.log(livre.afficher());
```